

障害時に複製したサービスにおける復旧後のデータ統合に関する研究

著者	大坂 優輝
雑誌名	東北大学電通談話会記録
巻	90
号	1
ページ	232-233
発行年	2021-08-20
URL	http://hdl.handle.net/10097/00132895

修士学位論文要約（令和3年3月）

障害時に複製したサービスにおける復旧後のデータ統合に関する研究

大坂 優輝

指導教員：長谷川 剛， 学位論文指導教員：北形 元

A Study on Data Integration of Replicated Network Services after Long-Term Disconnected Operation

Yuki OSAKA

Supervisor: Go HASEGAWA, Research Advisor: Gen KITAGATA

In the event of a network disruption, it is possible to restore and run services on each isolated network by storing the services' snapshots in a distributed manner during normal periods. However, it is difficult to avoid the inconsistency of the data updated in each isolated service when the services are merged into one after long-term operation. In this article, we discuss the data inconsistencies that occur when integrating services after long-term isolated operation. Then we propose a data management method that avoids these inconsistencies and enables integration. We show the effectiveness of the proposed method through experiments using a prototype system.

1. はじめに

ネットワーク障害が発生した際にネットワークサービスの可用性を向上させる手法として、サービス複製が考えられる。しかし、図1に示すように、サーバー間が相互に通信できない状況下で、複製サービスを長期的に並行して運用した場合（以下、長期的な孤立運用とする）、既存のデータベース管理手法ではサービスの持つデータを不整合無く統合することはできない^[1-3]。

そこで本稿では、データの不整合について意味的な考察を行い、サービスの長期的な孤立運用後に、データを不整合無く統合可能とするデータ管理機構を提案する。また、提案機構を取り入れた安否確認サービスの実装を行い、不整合が発生する入力を与えて提案機構の評価を行う。

2. データ統合の際の不整合

データの不整合は、データベースレベルの不整合と、アプリケーションレベルの不整合に分けられる。データベースレベルの不整合とは、トランザクションの一貫性が満たされないことを指す。また、アプリケーシ

ョンレベルの不整合とは、トランザクションの結果、意味的に正しいデータベースの状態から乖離し、サービスの運用に不都合が生じることを指す。

データベースレベルの不整合としては、更新喪失、ダーティリード、非再現リード、ファントムリードが挙げられる^[4]。これらの不整合はトランザクションの同時実行を許すことにより生じる。本稿では、異なる複数のサーバーにおいて同じタイミングに実行された複数のトランザクションがある場合でも、統合の際にそれらを同時実行せず、すべて直列に実行することにより、データベースレベルの不整合を回避する。

また、データベースレベルの不整合ではないが、アプリケーションレベルの不整合になりうる現象として、直列的更新喪失、重複追加、虚存更新、識別齟齬が考えられる。まず、直列的更新喪失は、あるサーバーで発生したトランザクションにより行われた更新の内容が、他のサーバーで発生したトランザクションにより行われた更新により上書きされ、先に行われた更新の内容が失われる現象である。次に、重複追加は、唯一性のあるデータが、複数のレコードとして INSERT される現象である。続いて、虚存更新は、DELETE が行われたレコードに対して UPDATE を行う現象である。最後に、識別齟齬は、シーケンス ID をデータベースのプライマリーキーとして用いた場合に、データ統合時にシーケンス ID が変更されることにより、操作を実行すべき対象のレコードが変化する現象である。

これらの不整合に関して考察をすると、長期的な孤立運用後に不整合無くデータを統合することができな

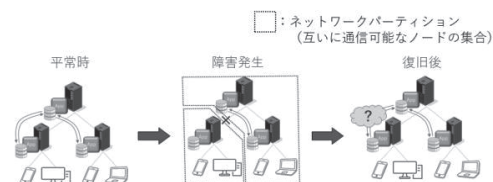


図1 想定する長期的な孤立運用のシナリオ

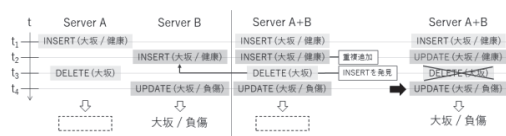


図2 重複追加，虚存更新の例と解決法

いサービスの条件は、以下の3つにまとめられる。

- ・各サーバーのデータをすべて維持する必要がある。
- ・トランザクションの順序を変更すると結果が変わる。
- ・識別に使用できるフィールドが無いレコードを扱う。

これらの条件に該当しないサービスであれば、長期的な孤立運用後にデータ統合が可能である。また、統合には以下の4つの操作が可能である必要がある。

- ・INSERT 実行前に当該レコードの有無を調べ、既存の場合は INSERT を UPDATE に置き換える。
- ・DELETE 実行前に、DELETE 実行日時より以前に他の全サーバーで実行された、同じものを指すレコードへの書込操作について検索し、直近に INSERT か UPDATE が実行されているサーバーが存在した場合には、DELETE を実行しない。
- ・シーケンス ID 以外をプライマリキーとして用いる。
- ・プライマリキーの値を変更する UPDATE を行う際に、他サーバーでの同じものを指すレコードに対する操作の、プライマリキーの値を新しい値に置き換える。

例えば、図 2 に示す重複追加と虚存更新の場合、2 度目の INSERT を UPDATE に置き換え、他サーバーで INSERT をしていることから DELETE は実行しないことにより、不整合を生じずにデータを統合できる。

3. データ管理機構の提案

図 3 に, 長期的な孤立運用後のデータ統合を実現するデータ管理機構の構造を示す. 本稿では,

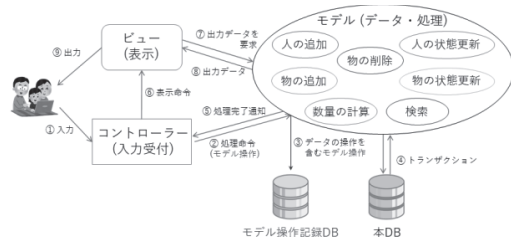


図3 提案機構の構造

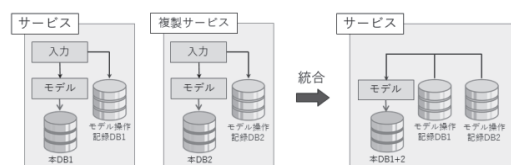


図4 提案機構によるデータ統合

```
private void addUserInputs1()
{
    userInput.Add(new UserInput(Action.Regist, new User("Sato Ichiro", "injured", null, 0)));
    userInput.Add(new UserInput(Action.Regist, new User("Suzuki Jiro", "injured", null, 1)));
}

private void addUserInputs2()
{
    userInput.Add(new UserInput(Action.Regist, new User("Sato Ichiro", "safe", null, 2)));
    userInput.Add(new UserInput(Action.Regist, new User("Takahashi Saburo", "safe", null, 3)));
}
```

id	name	status
1	Sato Ichiro	safe
2	Suzuki Jiro	injured
3	Takahashi Saburo	safe

図5 提案機構の評価実験

MVC モデル⁵⁾における, Controller から Model に対して送られる命令を記録するコンポーネントをサービスに組み込むことを提案する。以下, この命令をモデル操作と呼び, モデル操作を記録するコンポーネントをモデル操作記録データベースと呼ぶ。図 4 に提案機構を用いたデータ統合の様子を示す。複製サービスの長期的な孤立運用の後, それらのデータを統合する際には, 各複製サービスが持つモデル操作記録データベースからモデル操作を読み出して実行することで, ネットワーク障害が発生しなかったと仮定した際に想定されるモデルの挙動を再現する。

4. 提案機構の評価

提案機構を導入した安否確認サービスを実装し、不整合が生じる入力を与えて評価を行った。データベースのサーバーとして、Ubuntu 20.04 LTS 64 bit を使い、MySQL で操作を行った。また、プログラムの実装と実行はすべて Windows10 64 bit 上の Eclipse で行い、実装言語は Java を用いた。図 5 に入力と統合結果を示す。重複追加を生じる入力を与えたところ、統合結果に不整合は生じなかった。したがって、提案機構が長期的な孤立運用後の統合に対し有効であることが示された。

5. まとめ

本稿では、データ統合の際に生じる不整合についての考察と、考察をもとにした長期的な孤立運用後のデータ統合を可能とするデータ管理機構の提案を行った。提案機構を導入した評価用サービスを実装し、不整合が生じる入力を与え、統合結果に不整合が生じないことを確認し、提案機構の有効性を示した。

文献

- 1) N. Schiper, R. Schmidt, and F. Pedone, Proc. 10th Inte. Conf. Prin. Dist. Syst., 81–93, (2006).
- 2) D. Sciascia, F. Pedone, and F. Junqueira, Proc. Inte. Conf. Depe. Syst. Netw., 1–12, (2012).
- 3) P.T. Wojciechowski, T. Kobus, and M. Kokocinski, Trans. Para. Dist. Syst., 28(3), 891–904, (2017).
- 4) 白鳥則郎ら, データベース —ビッグデータ時代の基礎—, 共立出版, (2018).
- 5) G.E. Krasner, S.T. Pope, et al., J. Obje. Orie. Prog., 1(3), 26–49, (1988).